CISC 3115 MY3

Programming Environment and Essential Tools

Hui Chen

Department of Computer & Information Science

CUNY Brooklyn College

Outline

- Get around in Operating Systems
  - Unix-like: Unix, Linux, OS X and Windows
  - Terminal and command line
- Development environment
  - Set up program authoring tools
    - Editors
  - Set up JRE and JDK

Programming …

"*Computer science is no more about computers than astronomy is about telescopes*"

-- Edsger Dijkstra

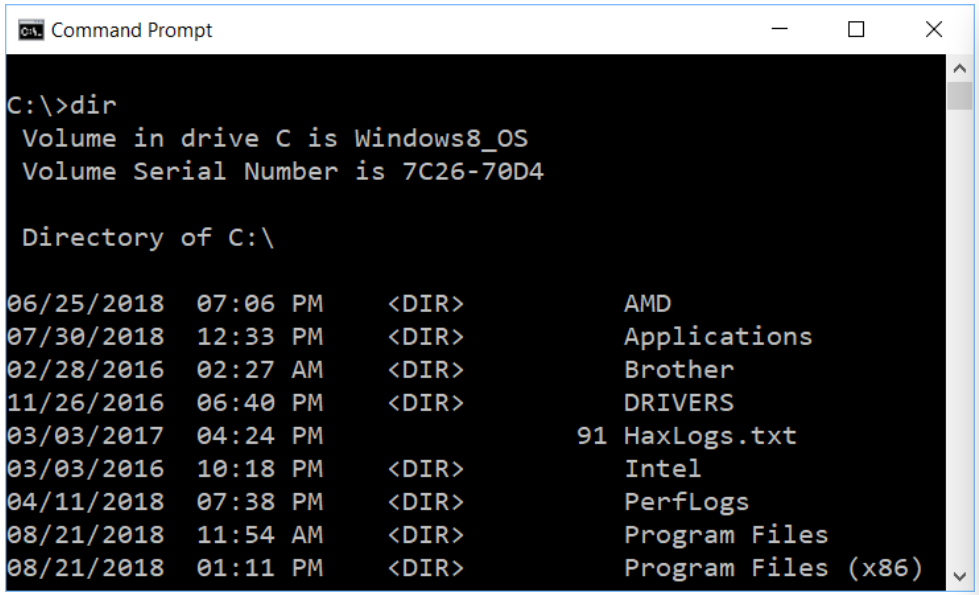
- Our objective as a computer scientist:
  - To solve computational problems
  - By developing computer programs
  - On computer systems
- We shall have a brief discussion about Operating Systems and programming environment

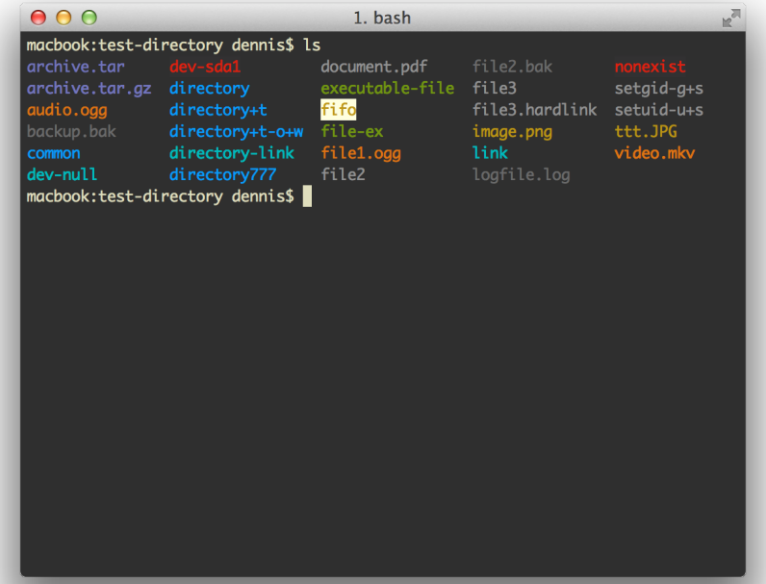
Operating Systems

- Do you know the answers to the questions:
  - Where are my files?
  - What is a folder or a directory?
  - How do I install an computer application?
  - How do I open a terminal window?
  - How do I copy/delete/rename a file, or a folder/directory. How about several files or folders/directories?
- To learn programming, it is necessary to be proficient in using your computer. Minimally, you need to know the answers to the questions like these.
  - If not yet, no sweat, we will get there this semester.

Terminal and Command Line

- Use a Command Line
- Why?





Common Tasks on Command Line

Windows

- Display working directory: cd
- Display content of a directory: dir
- Change directory: cd directory_to_go
- Make directory: mkdir dir_to_make
- Move files: move this_file   to_dir
- Delete file: del file_to_delete
- Delete directory: rmdir dir_to_delete

Unix-like

- Display working directory: pwd
- Display content of a directory: ls
- Change directory: cd directory_to_go
- Make directory: mkdir dir_to_make
- Move files: move this_file   to_dir
- Delete file: rm file_to_delete
- Delete directory: rmdir dir_to_delete

Questions?

- Can you get around on Windows/Unix-like systems using command line?

JRE and JDK

- JRE: Java Runtime Environment
  - Required to run a Java program
- JDK: Java Development Kit
  - Required to compile and run a Java program
- Have you had JDK installed/set up in your system?

Oracle JDK vs Open JDK

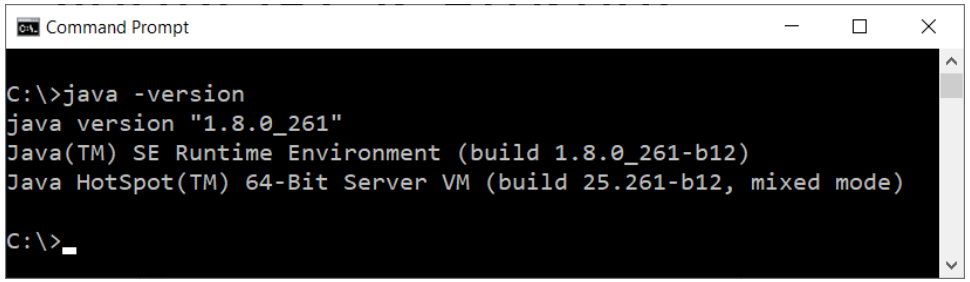
- Java
  - A specification. There are more than one implementations and packaging.
  - https://docs.oracle.com/javase/specs/index.html
- Oracle JDK
  - https://www.oracle.com/java/technologies/javase-downloads.html
- Open JDK
  - https://openjdk.java.net/
- But, wait!!!

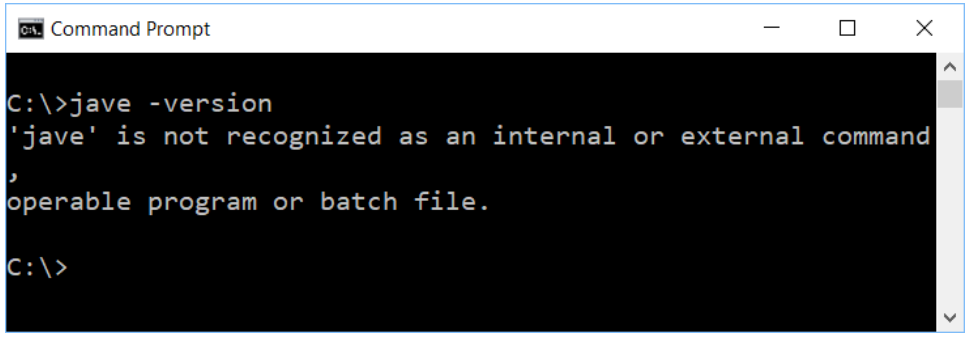
Java Version

- For this class, Java 11 LTS or 17 LTS or 21 LTS is required.
  - https://www.oracle.com/java/technologies/javase/jdk11-archive-downloads.html
  - https://www.oracle.com/java/technologies/javase/jdk17-archive-downloads.html
  - https://www.oracle.com/java/technologies/javase/jdk21-archive-downloads.html

Verify JRE is Present

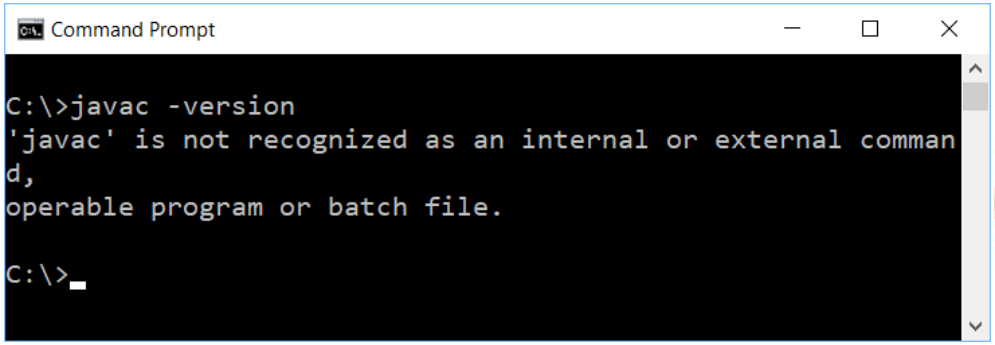
- On Command Line
  - java -version

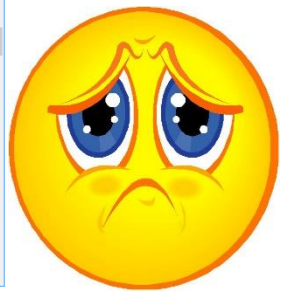




Verify JDK is Present

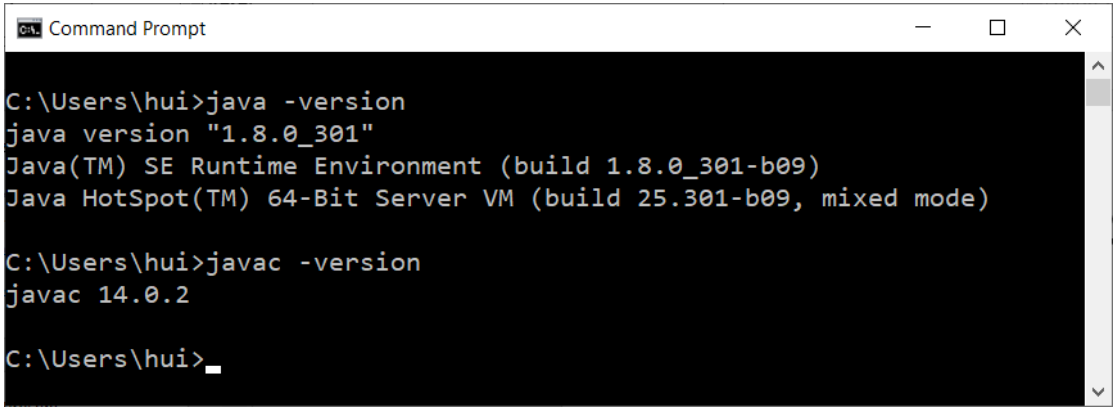
- On Command Line (screen shots are dated)
  - javac -version

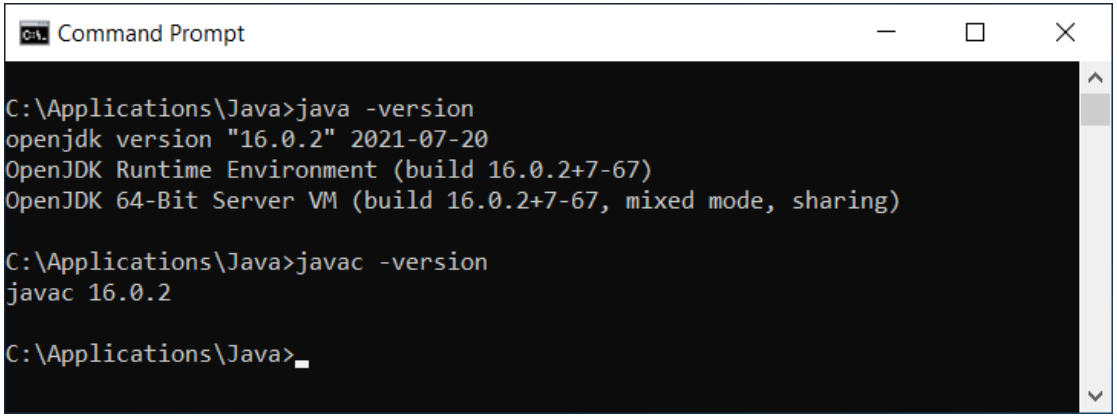




Verify Versions of JVM and Java Compiler

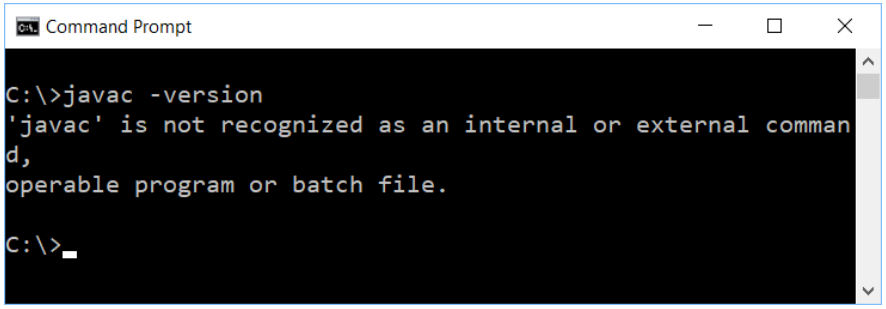
- On command line (screen shots are dated)





Trouble with JDK?

- "I have installed JDK, but still I got this!"



- You need to set up the search path for JDK's executables (such as, javac)
  - System specific, but we will do it via the user profile of "Git Bash"

In-Class Exercise

- Let’s complete the following tasks
  - Open a terminal
  - Show working directory
  - Display content in the working directory
  - Switch to a different directory
  - Create a directory
  - Delete the directory
  - Verify if JRE is present
  - Verify if JDK is present (if inaccessible on command line, it is OK for this exercise; we shall address this in a little while)
  - Write a short manual about the steps in the journal in directory C0826

In-Class Exercise

1. Create a folder (i.e., directory) , such as “CISC3115” somewhere on your computer
2. In the “CISC3115” directory, create a subfolder (i.e., subdirectory), e.g., for today, create the “C0829” folder.
3. Go to the directory (“cd C0829”)
4. Create a batch file (on Windows systems)/shell script (on UNIX systems)
   - Windows

     echo java -version > checkjava.cmd

     echo javac -version > checkjavac.cmd
   - UNIX

     echo “java -version“ > checkjava.sh

     echo “javac -version“ > checkjavac.sh

Questions?

- Can you either
  - Set up a required version of JDK, or
  - Verify it has been already set up?

Coding Environment for the Class

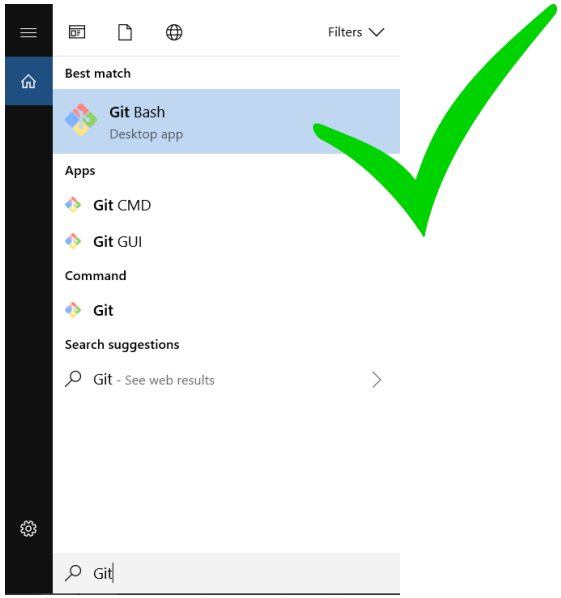
- Instructor's preferred development environment during lecture demos
  - Git Bash + Atom editor + JDK
- Online IDE
  - Examples: replit.com; Github codespaces
- (Optional, not recommended) Using Desktop IDEs
  - IntelliJ IDEA
  - Eclipse
  - NetBeans
  - BlueJ
  - Dr. Java
  - …
- https://www.sci.brooklyn.cuny.edu/~goetz/java/

Prepare Coding Environment

1. Install the git client (if not already installed)
   - Why do we need this?
     - For Windows users, use Unix-like terminal on Windows
     - Version your submitted work
     - Get feedback from autograder (provided by the instructor)
2. Install the Atom editor (if not already installed)
   - Why do we need this?
     - This is the instructor's choice to author simple Java programs
     - It has syntax highlighting, but limited autocomplete and other features, which forces us to learn the language and the API.

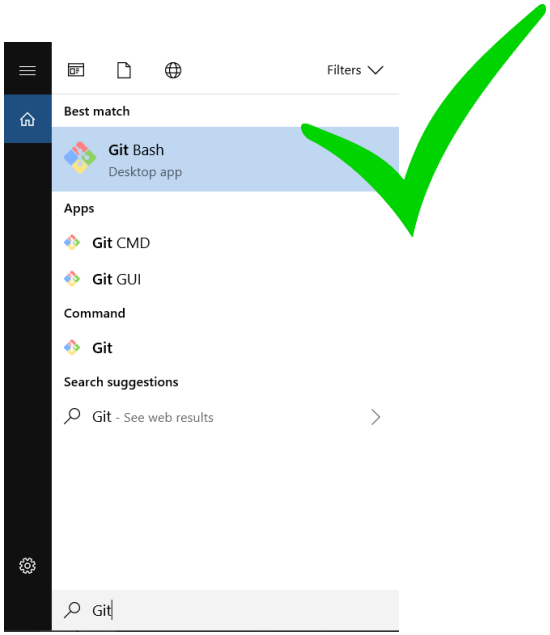
Verify Whether You Have Git Client

- Verify if you have had the Git client installed already
- Windows
  - Attempt to run “Git Bash”
- Unix (OS X or Linux):
  - Open a terminal window
  - Run “git --version”, i.e., type “git --version“ (without quotes) and hit the Enter key



Have I Had Git Client Installed?

- Windows and Unix

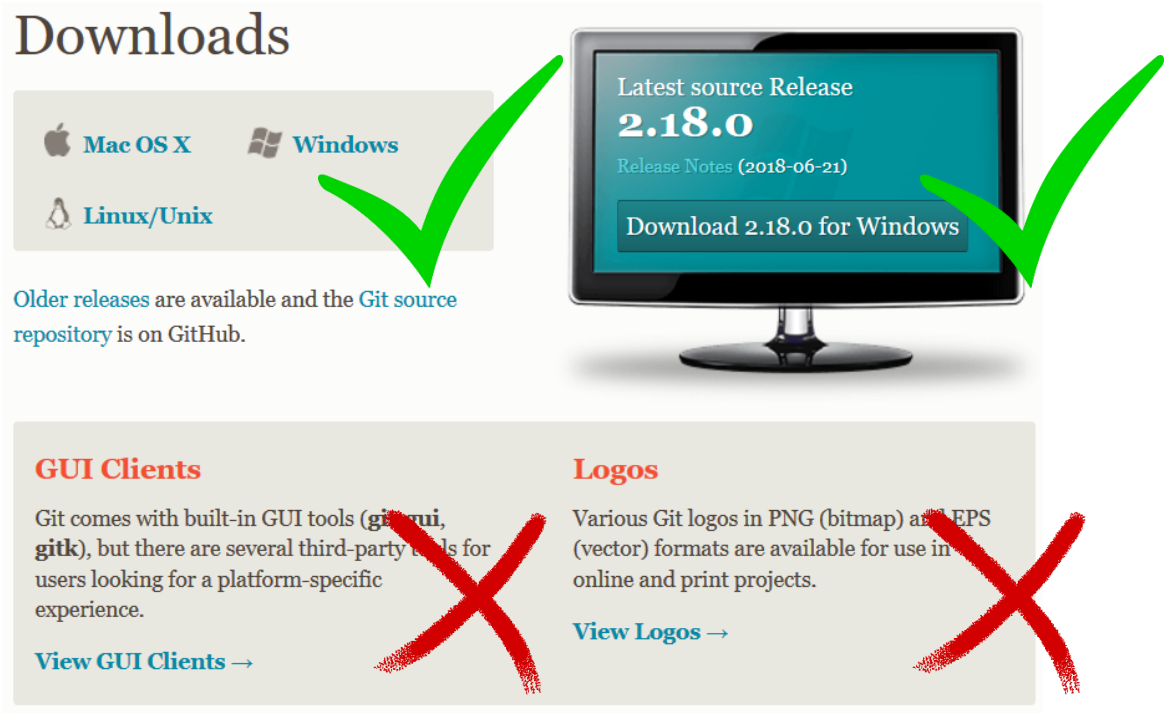



```
$ git --version
git version 2.17.1
```

- If not, download and install it

Download Git Client

- Visit https://git-scm.com/downloads using your favorite Web browser

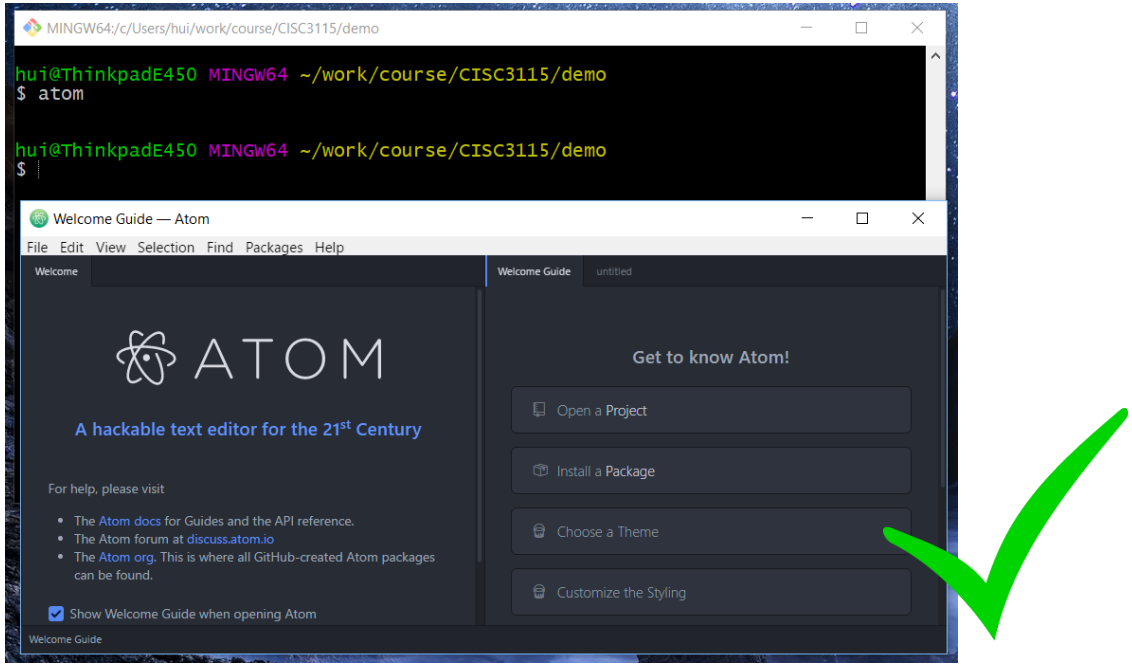


Git Bash on Windows

- Provides a terminal where you can run Unix commands
- The instructor shall use the Git Bash from now on so that the instructions are identical to both Windows and Unix (e.g., OS X) users
- Window users: Use the Git Bash terminal
- Unix users: just use a terminal (e.g., the terminal on OS X)

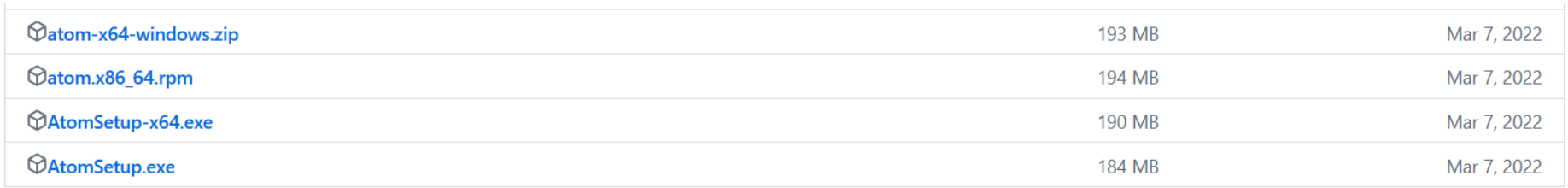
Verify Whether You Have Atom Installed

- Verify if you have had the Atom editor installed already
  - Type atom on the Command Line



Download and Install the Atom Editor

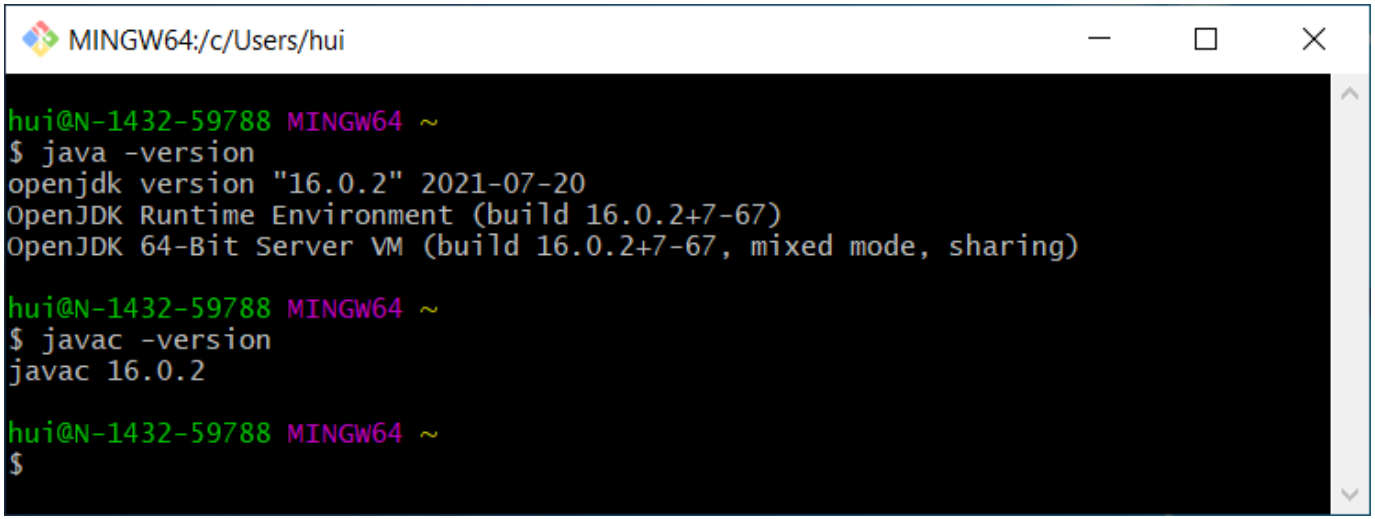
- If you have not had the Atom Editor installed, download and install the Atom editor
- Visit
  - https://github.com/atom/atom/releases/tag/v1.60.0



atom-x64-windows.zip	193 MB	Mar 7, 2022
atom.x86_64.rpm	194 MB	Mar 7, 2022
AtomSetup-x64.exe	190 MB	Mar 7, 2022
AtomSetup.exe	184 MB	Mar 7, 2022

Checking on Java and Javac

- Check whether both java & javac are found, and have an identical version. Otherwise, next slide.

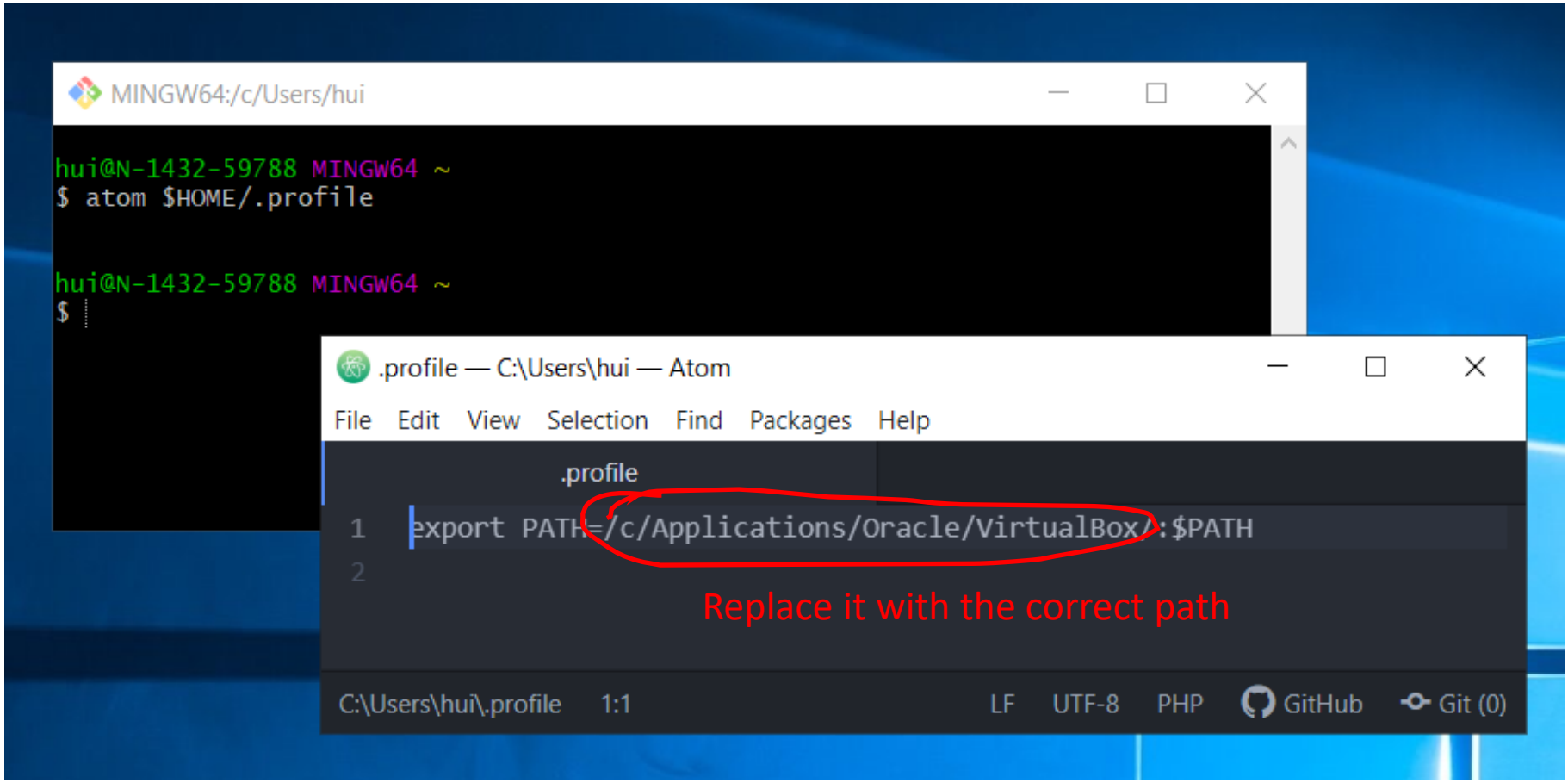


- Based on the screenshot, did I have the required version of JDK set up?

Setting up Search Path for java and javac

- Sometimes we also need to do this step:
  - In “Git Bash” terminal, create (if not alrady exists) or edit the .profile file on your “home directory” (see next slide)
  - Then, restart “Git Bash” terminal, and check accessibility and versions of Java and Javac

Edit/Create .profile File



Questions?

- Can you set up git or verify the existence of git on your systems?
- Can you set up atom on your system?